11-4-2020

**PROYECTOS INFORMÁTICOS II**

**MEMORIA DEL PROYECTO**PLANIFICACIÓN Y ANÁLISIS GENERAL

**MONOPOLY**

GRUPO A

DANIEL – AYOUB – RAÚL

**ÍNDICE**

[1. Control de documentación 2](#_Toc38785398)

[2. Historial de modificaciones 2](#_Toc38785399)

[3. Ciclo de vida del proyecto 3](#_Toc38785400)

[3.1 CICLO DE VIDA DE SOFTWARE 4](#_Toc38785401)

[4. Fase de planificación del proyecto 6](#_Toc38785402)

[4.1 Planificación en fase de ejecución e hitos 7](#_Toc38785403)

[4.2 Equipo del proyecto 7](#_Toc38785404)

[4.3 Análisis DAFO 8](#_Toc38785405)

[4.4 Casos de uso 9](#_Toc38785406)

[4.5 Modelo Objeto-Relación 10](#_Toc38785407)

[4.6 Diagramas de actividad 11](#_Toc38785408)

[4.7 Diagramas de secuencia 17](#_Toc38785409)

# Control de documentación

|  |  |  |
| --- | --- | --- |
| **Elaborado por:** | **Revisado por:** | **Aprobado por:** |
| Alumnos de la asignatura (desarrollo) | Alumnos de la asignatura (revisado) | Equipo de la Universidad |
| **Nombre:** | **Nombre:** | **Nombre:** |
| Daniel Ortega  Ayoub Dachour  Raúl García | Daniel Ortega  Ayoub Dachour  Raúl García | [Jordi José](https://opencampus.uols.org/user/view.php?id=1347&course=872) |
| **Fecha:** | **Fecha:** | **Fecha:** |
| 08/05/2020 |  |  |
| **Firma:** | **Firma:** | **Firma:** |
|  |  |  |

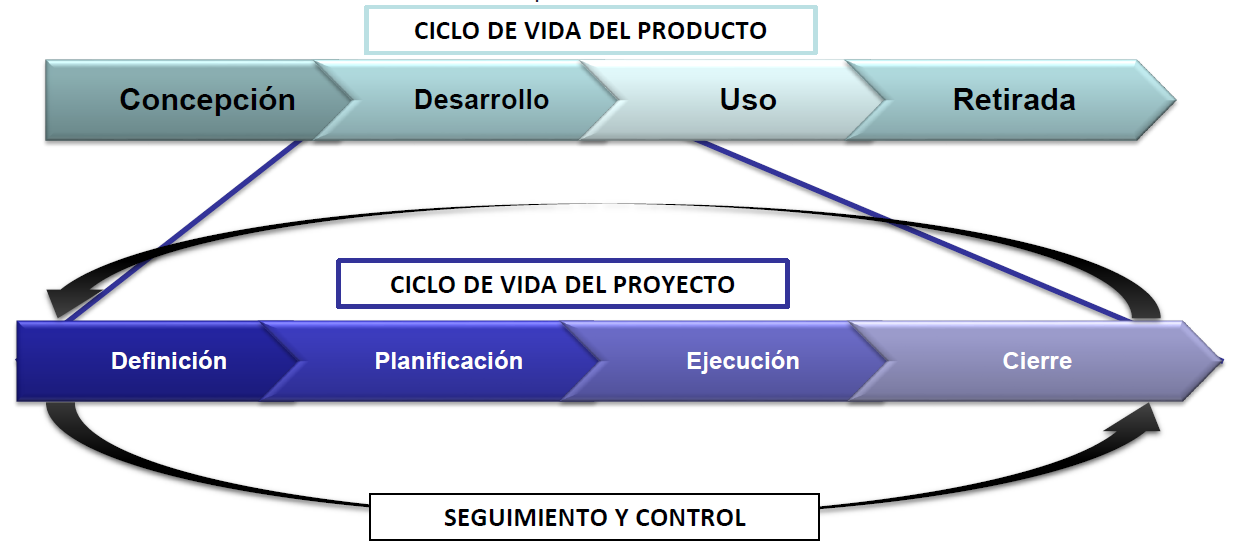
# Historial de modificaciones

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSIÓN** | **FECHA DE VIGENCIA** | **MODIFICADO POR:** | **MODIFICACIÓN REALIZADA** |
| 1.0 | 27/03/2020 | Daniel Ortega | Creación de documento |
| 1.1 | 27/03/2020 | Daniel Ortega | -Recuperar casos de uso y requisitos  -Planificación detallada del proyecto |
| 1.2 | 08/04/2020 | Raúl García | - Integrar partes que faltan de la tarea anterior  - Incluir diagrama de clases |
| 1.3 | 09/04/2020 | Raúl García | - Repaso apartado 3.  - Añadir apartado 6 “Fase de control del proyecto” |
| 1.4 | 10/04/2020 | Raúl García | - Ajuste formato  - Añadir contenido apartado 5. Descripción de clases  - Añadir contenido apartado 6 |
| 1.5 | 11/04/2020 | Raúl García | - Añadir contenido apartado 6 |
| 1.6 | 11/04/2020 | Daniel Ortega | - Ajuste formato  - Adaptar fase control y calidad |
| 1.7 | 25/04/2020 | Daniel Ortega | División de documento en tres. Este documento se enfoca a análisis general y planificación |

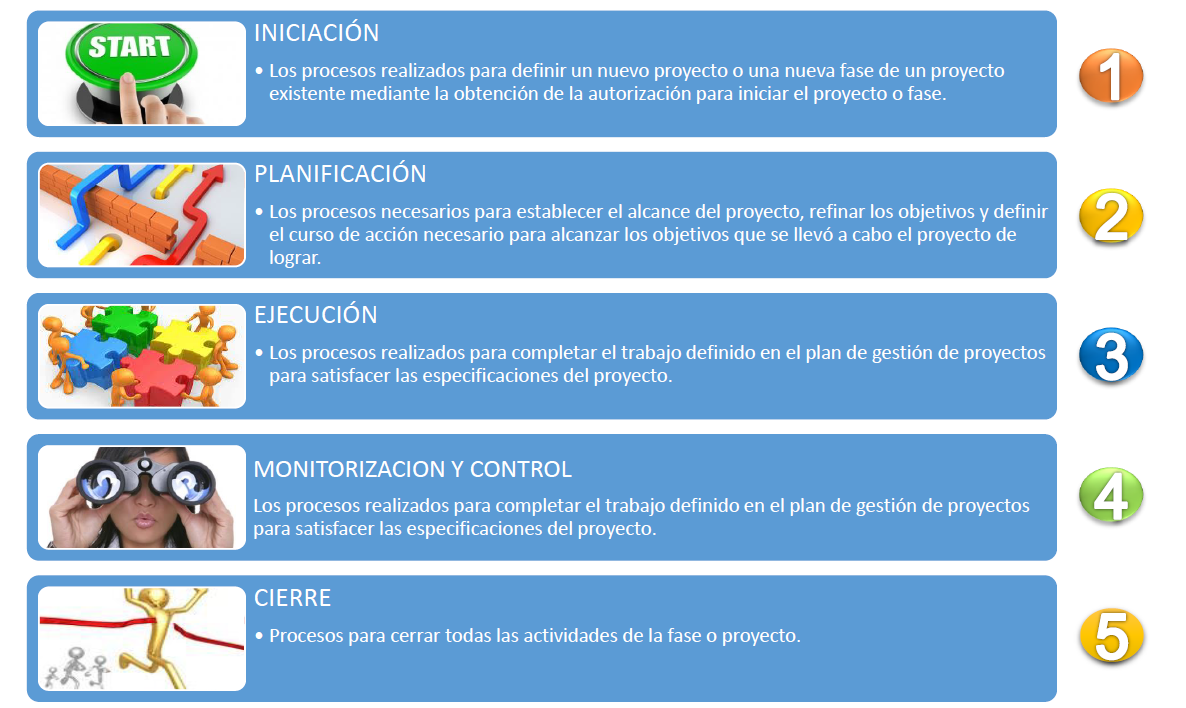
# Ciclo de vida del proyecto

Un proyecto está dividido en diferentes fases bien diferenciadas, el ciclo de vida del proyecto define las fases que conectan el inicio con el cierre del proyecto. Este ciclo de vida está compuesto por 4 fases, siendo la fase de ejecución transversal a la de control.

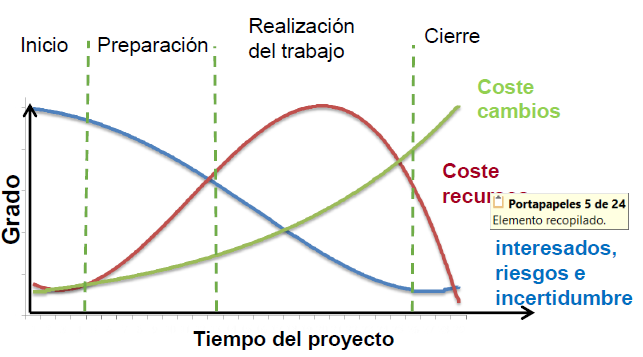
El objetivo de esta memoria de proyecto será recoger y garantizar que todas las tareas de cada fase en el ciclo de vida del proyecto se llevan a cabo. Si no es así, se deberá dejar registrado el porqué de cada caso.



Se adjunta a continuación una descripción breve de lo que implica cada fase de un proyecto, sea de la naturaleza que sea.



A continuación, se muestra un gráfico con tres variables que influyen en el desarrollo de un proyecto: coste cambios, coste recursos e incertidumbre. Como podemos ver el coste de recursos es mayor en la etapa de ejecución.



# CICLO DE VIDA DE SOFTWARE

Luego, hay un ciclo de vida del software, donde a continuación, lo describiremos como un sub-apartado, donde se describe desde la fase inicial hasta la fase final.

A continuación, se detallan las diversas fases:

* Recopilación y análisis de requisitos.
* Diseño
* Implementación o codificación
* Pruebas
* Despliegue
* Mantenimiento

1. Recopilación y análisis de requisitos

Durante esta fase, toda la información relevante se recopila del cliente para desarrollar un producto según sus expectativas. Cualquier ambigüedad debe resolverse solo en esta fase.

El analista de negocios y el Gerente de Proyecto deben establecer una reunión con el cliente para recopilar toda la información, como lo que el cliente quiere construir, quién será el usuario final, cuál es el propósito del producto. Antes de construir un producto, una comprensión o conocimiento básico del producto es muy importante.

Por ejemplo, un cliente quiere tener una aplicación que implique la realización de un juego como el Monopoly. En este caso, el requisito debe ser claro, como qué tipo de transacciones se realizarán, cómo se realizará, en qué modo, etc.

Una vez que se realiza la recopilación de requisitos, se realiza un análisis para verificar la viabilidad del desarrollo de un producto. En caso de ambigüedad, se establece una llamada para una reunión adicional.

Una vez que se comprende claramente el requisito, se crea el documento de requisitos de software. Este documento debe ser entendido por los desarrolladores y también debe ser revisado por el cliente para futuras referencias.

**2)** Diseño

En esta fase, el requisito reunido en el documento de requisitos se utiliza como una entrada y se deriva la arquitectura de software que se utiliza para implementar el desarrollo del sistema.

**3)** Implementación o codificación

La implementación / codificación comienza una vez que el desarrollador obtiene el documento de diseño. El diseño del software se traduce en código fuente. Todos los componentes del software se implementan en esta fase.

**4)** Pruebas

Las pruebas comienzan una vez que se completa la codificación y los módulos se liberan para la prueba. En esta fase, el software desarrollado se prueba exhaustivamente y los defectos encontrados se asignan a los desarrolladores para repararlos.

Re-testear, la prueba de regresión se realiza hasta el punto en que el software está según las expectativas del cliente. Los evaluadores remiten el documento de los requisitos para asegurarse de que el software cumpla con el estándar del cliente.

**5)** Despliegue

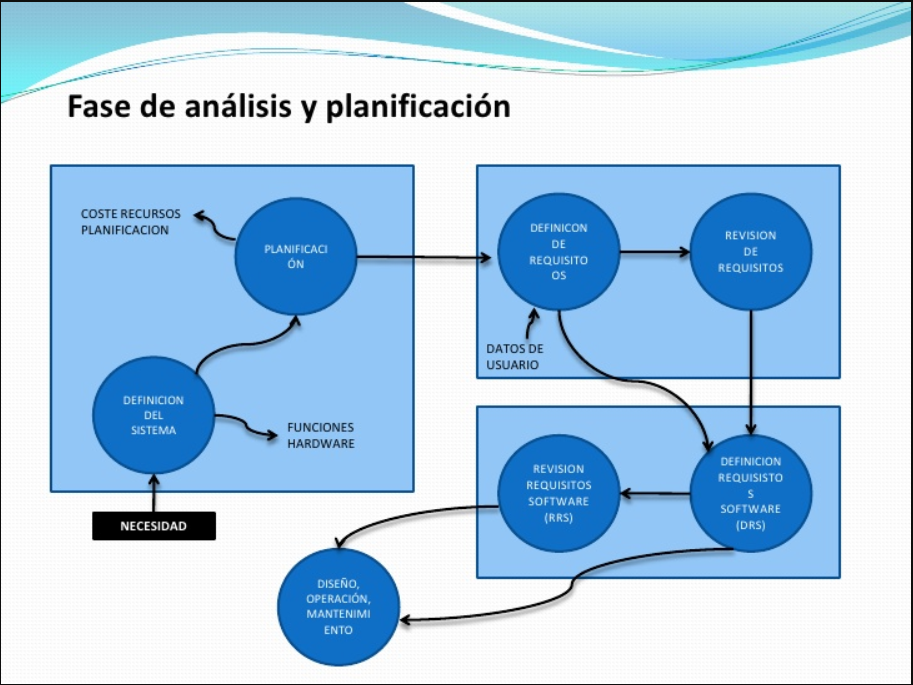
Una vez que se prueba el producto, se implementa en el entorno de producción y se realiza la primera prueba de aceptación del usuario según las expectativas del cliente.

**6)** Mantenimiento

Después del despliegue de un producto en el entorno de producción, los desarrolladores se ocupan del mantenimiento del producto, es decir, si surge algún problema y necesita ser reparado o si se debe realizar alguna mejora.

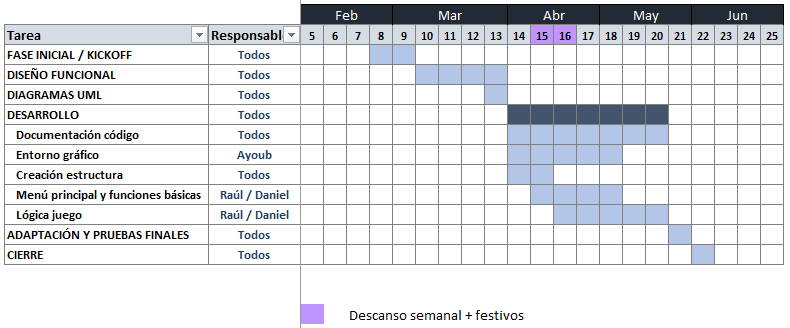
# Fase de planificación del proyecto

La fase del plan prepara el equipo para un rendimiento eficiente durante la ejecución. Es donde investigan y planifican el proyecto. Esta fase puede consistir en elaborar un plan de desarrollo del software, estableciendo las estimaciones del proyecto y creando un plan de aseguramiento de la calidad. También puede implicar el desarrollo de un plan de etapas de entregas, requerimientos, documentos de diseño detallado, un plan de gestión del cambio y gestión de riesgos. Además, la arquitectura del producto será acordada, el personal comenzará el aumento gradual y prototipos desarrollados.



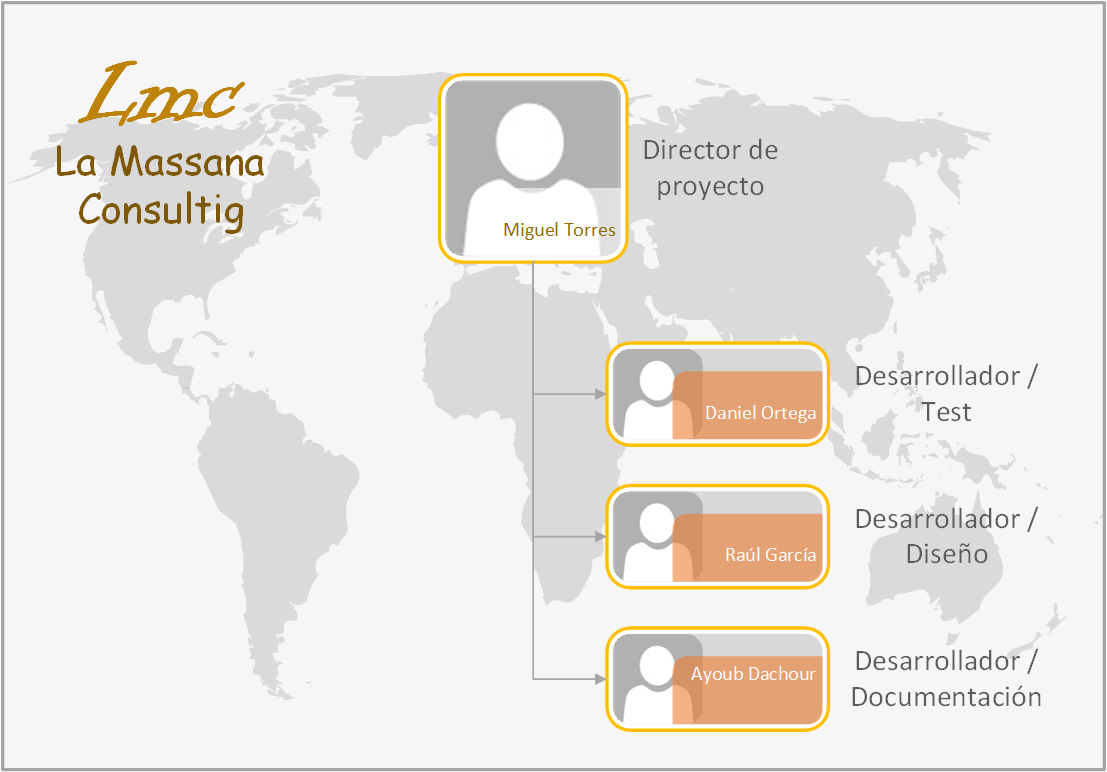
# Planificación en fase de ejecución e hitos

A continuación se detalla la planificación en detalle del proyecto que debe ser actualizada constantemente. Se divide por fases del desarrollo y casos de uso y las fases relacionadas.



# Equipo del proyecto

A continuación se incluye un diagrama con la jerarquía para este proyecto por parte de “LMC La Massana Consulting”, en adelante simplemente “LMC”.



El reparto de tareas queda mostrado en la planificación y diagrama de Gantt mostrado anteriormente:

|  |  |  |
| --- | --- | --- |
| Desarrollador | ID Equipo | Fase Ejecución / Control |
| Daniel Ortega | 1 | Memoria proyecto, desarrollo y test  Desarrollo |
| Ayoub Dachour | 2 | Memoria proyecto y entorno gráfico |
| Raúl García | 3 | Memoria proyecto, desarrollo y test |

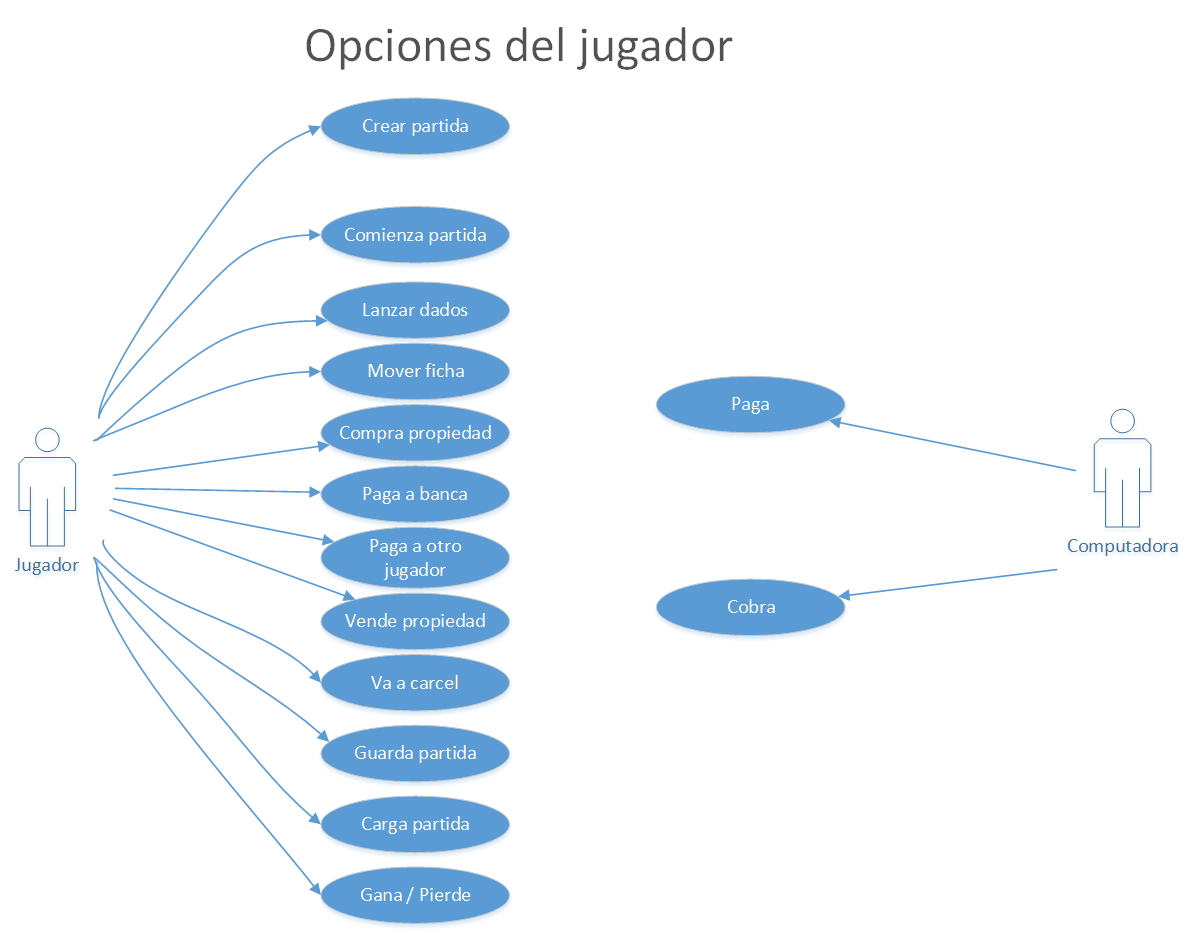
# Análisis DAFO

A continuación se expone la matriz DAFO para identificar debilidades, amenazas, fortalezas y oportunidades de la herramienta a realizar.

|  |  |
| --- | --- |
| **Debilidades**   * Poca experiencia desarrollando videojuegos en Java * Falta de formación en algunos aspectos como entorno gráfico * Sobrecarga de proyectos en el equipo | **Amenazas**   * Desarrollo no salga como esperado * Fallos en el código fuente * Test mal realizado * Perdida del cliente * Competencia muy alta * Requiere mucho tiempo de preparación y en paralelo hay que desarrollar |
| **Fortalezas**   * Mucha experiencia con otros lenguajes de programación * Equipo cualificado para hacer un videojuego a la altura de las expectativas * Promueve el desarrollo personal del equipo | **Oportunidades**   * Captar la atención del cliente y conseguir más proyectos * Conseguir nuevos clientes mediante recomendación del propio cliente * Adquirir experiencia en metodologías agiles * Mejora conocimiento IDE Netbeans y Framework Swing |

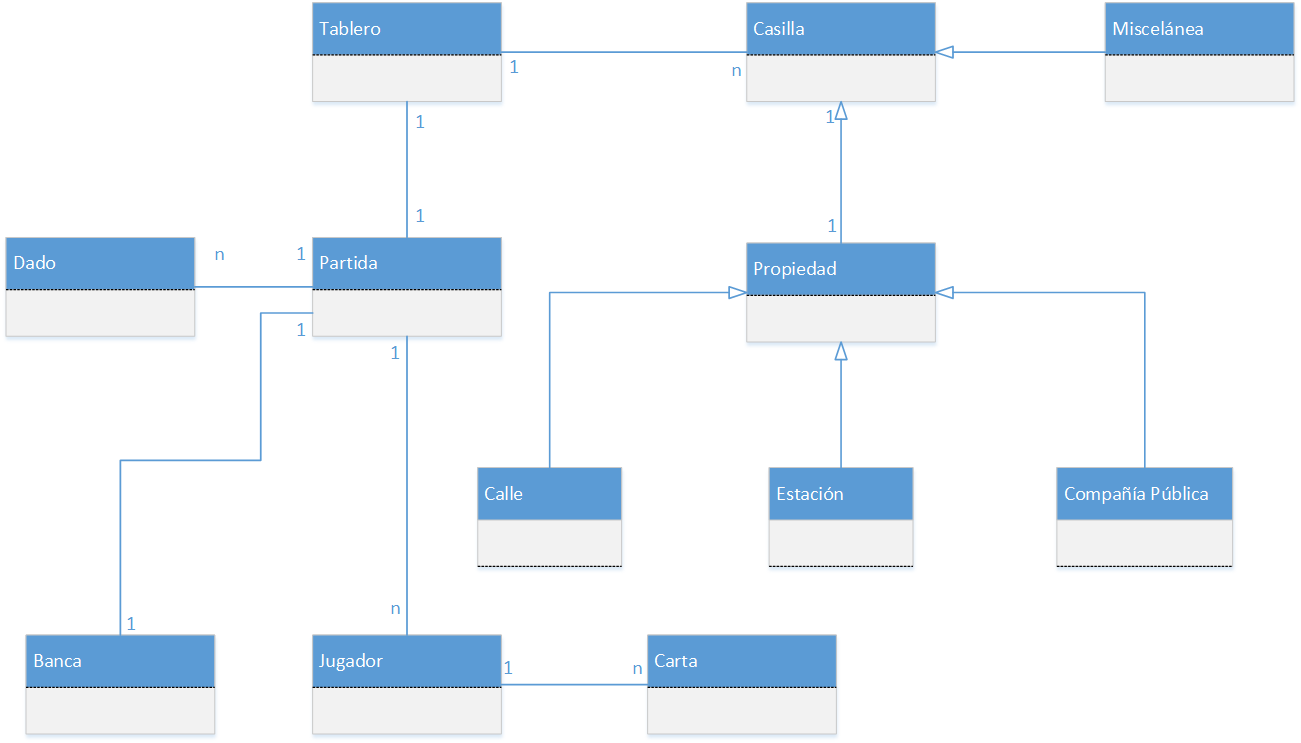
# Casos de uso

A continuación se expone casos de uso genéricos, con las principales funcionalidades que dispondrá la aplicación. En el futuro se estudiará realizar una evolución/modificación de estos e implementación de diagrama de clases.



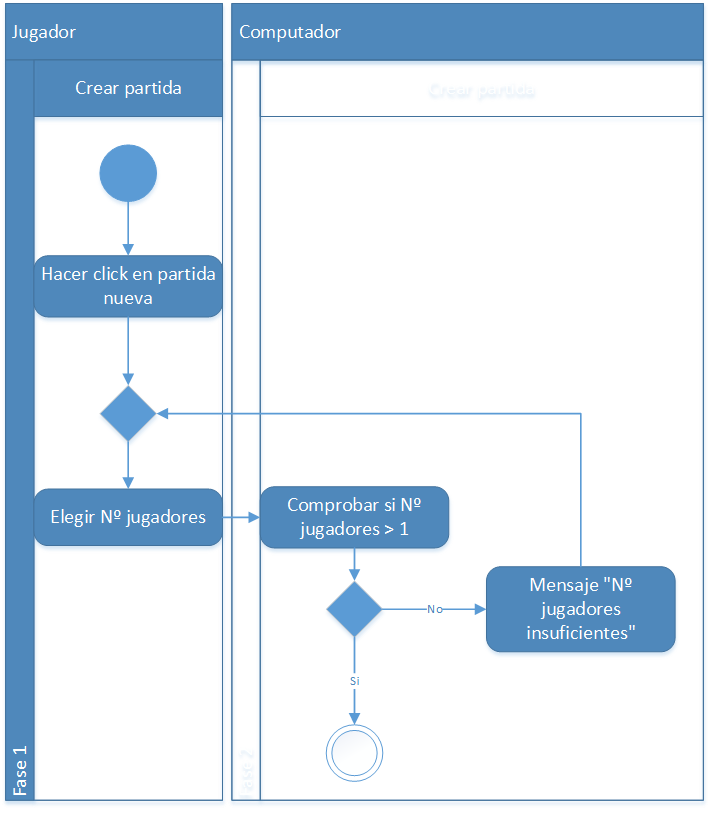
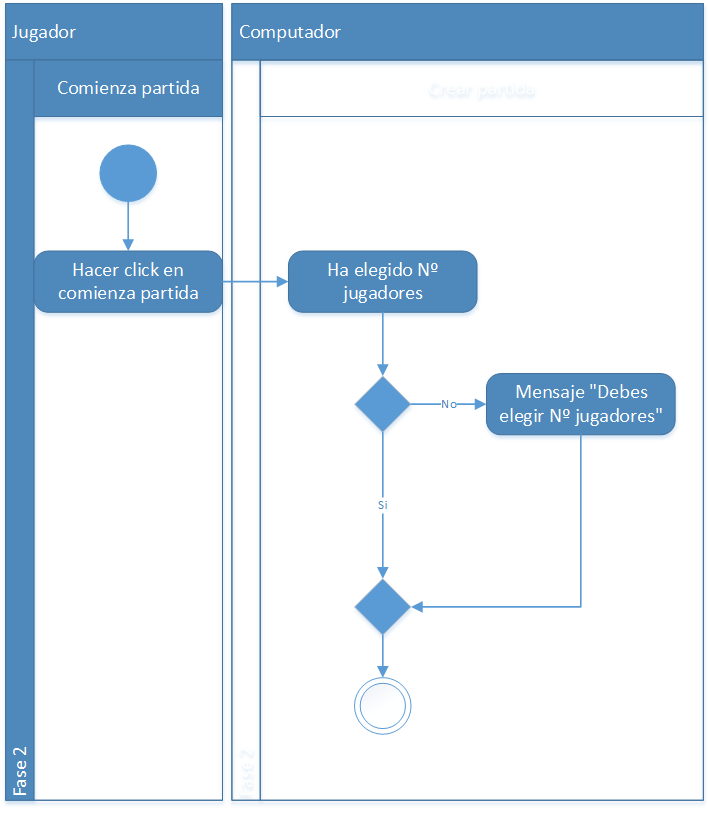
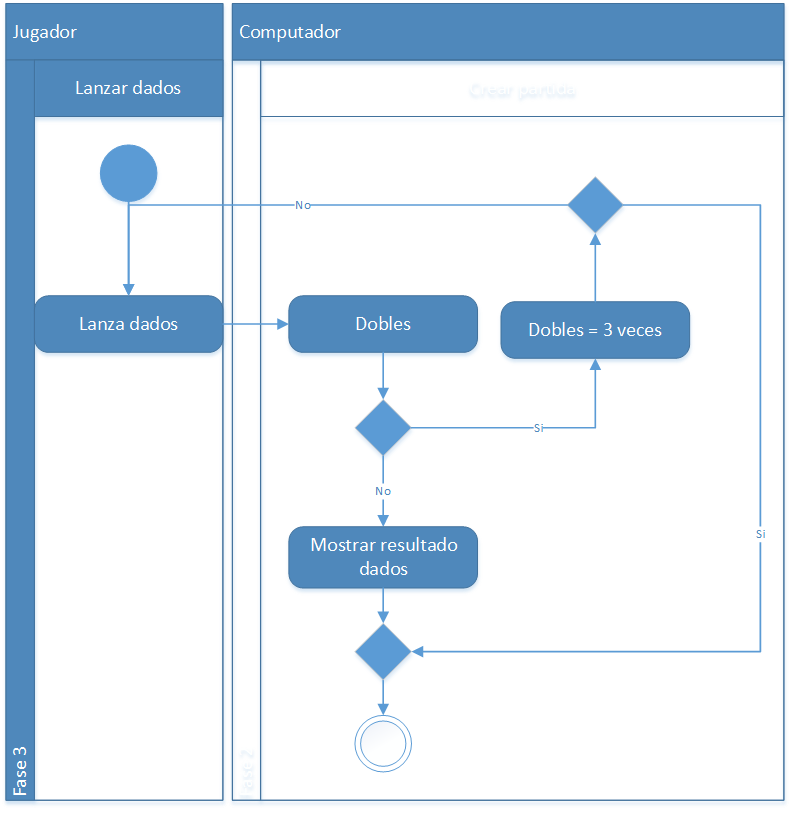
# Modelo Objeto-Relación

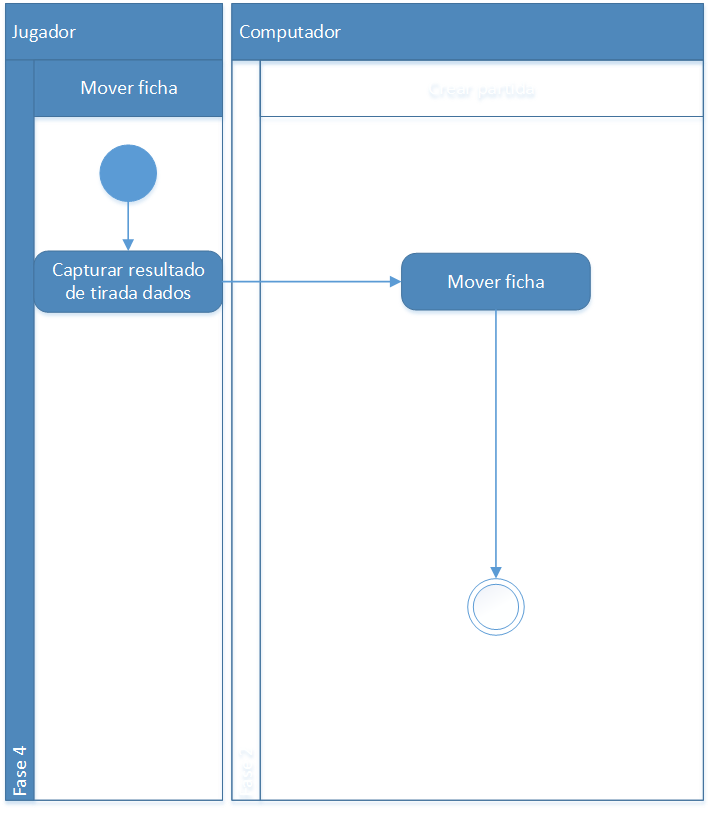
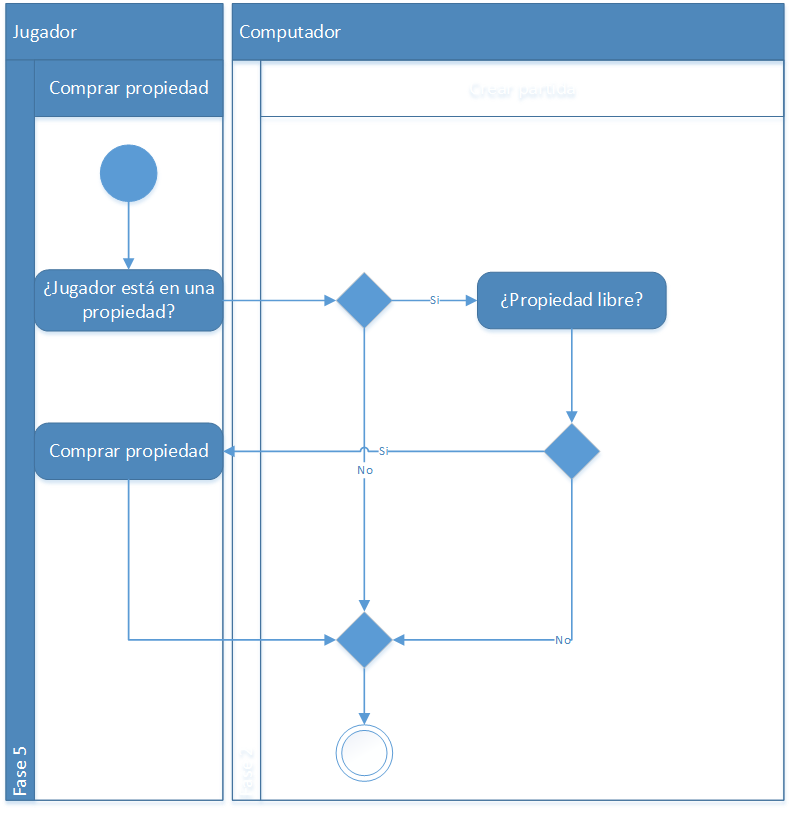
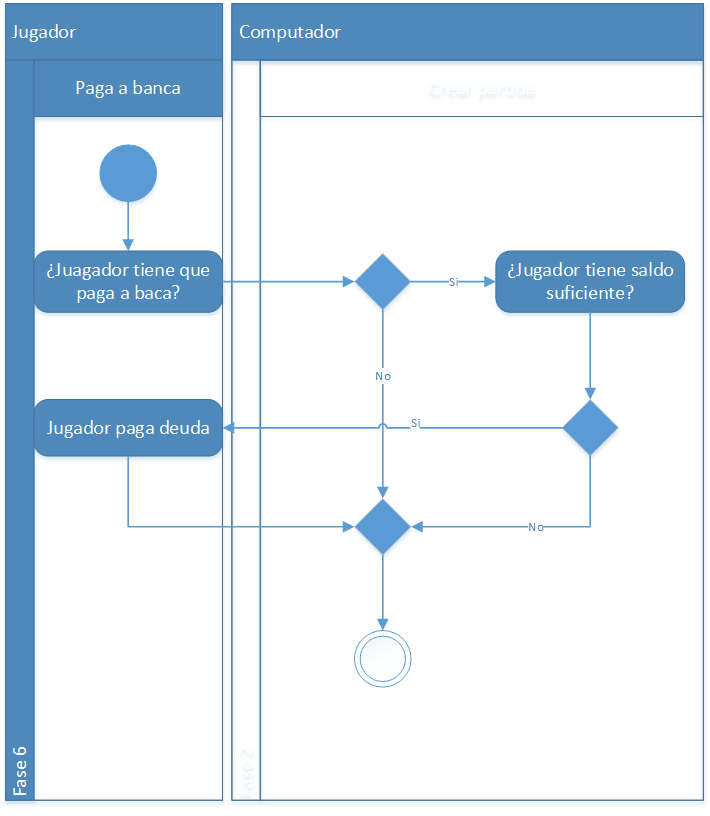
En este apartado vamos a exponer el modelo que, aunque no implementaremos ninguna base de datos, nos ayudará a comprender la relación entre objetos de la aplicación. Este modelo es una aproximación a alto nivel que puede sufrir cambios a lo largo del desarrollo del proyecto.

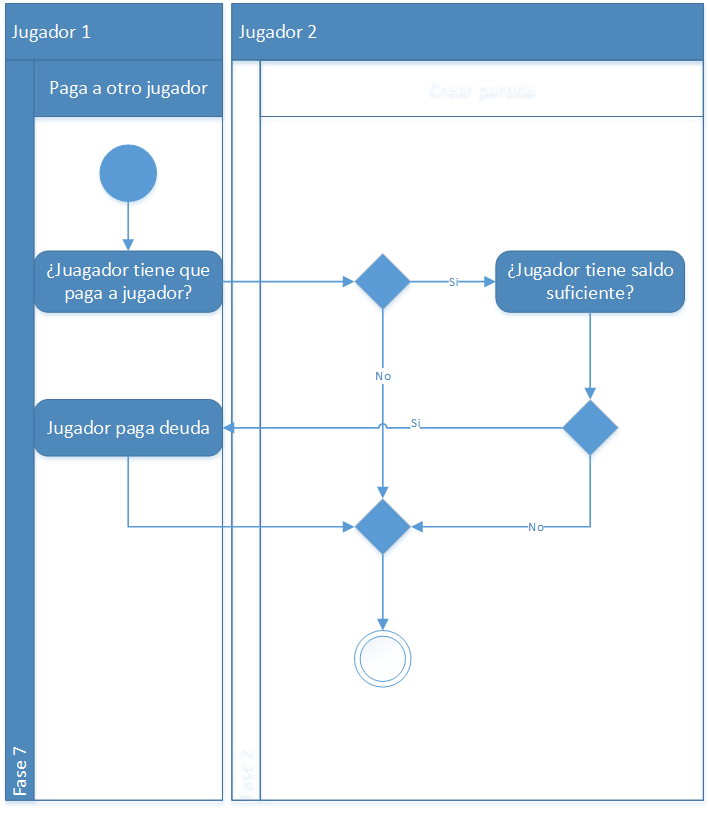
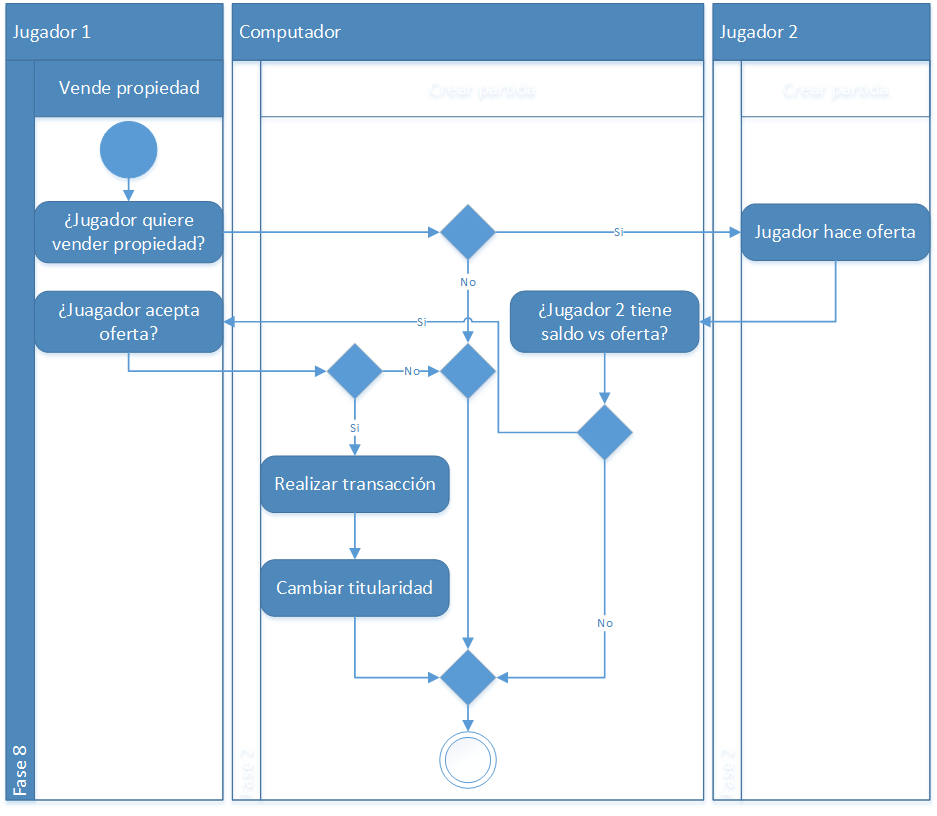


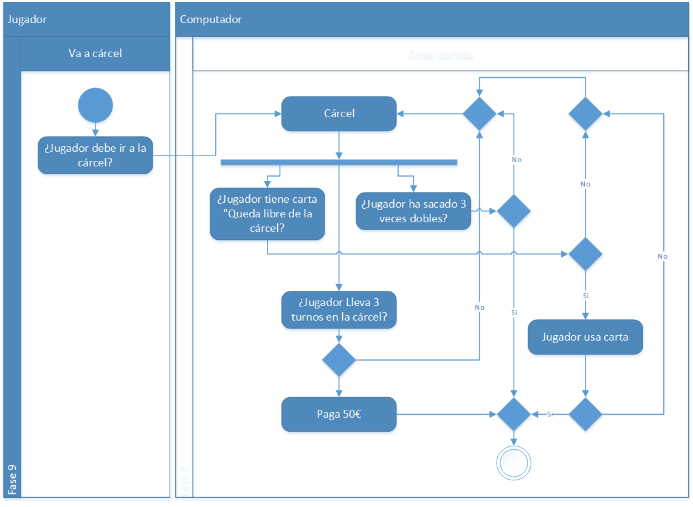
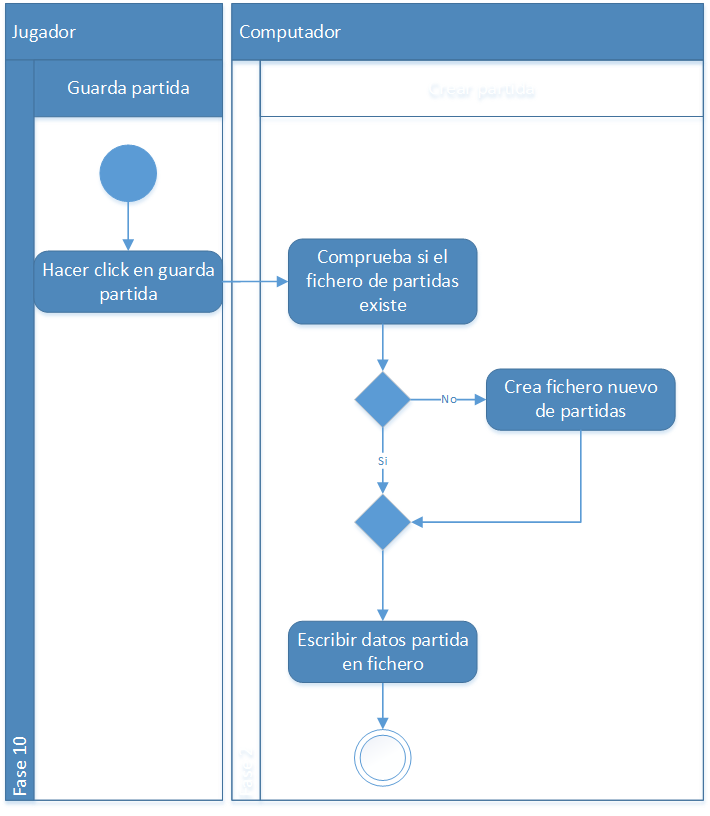
# Diagramas de actividad

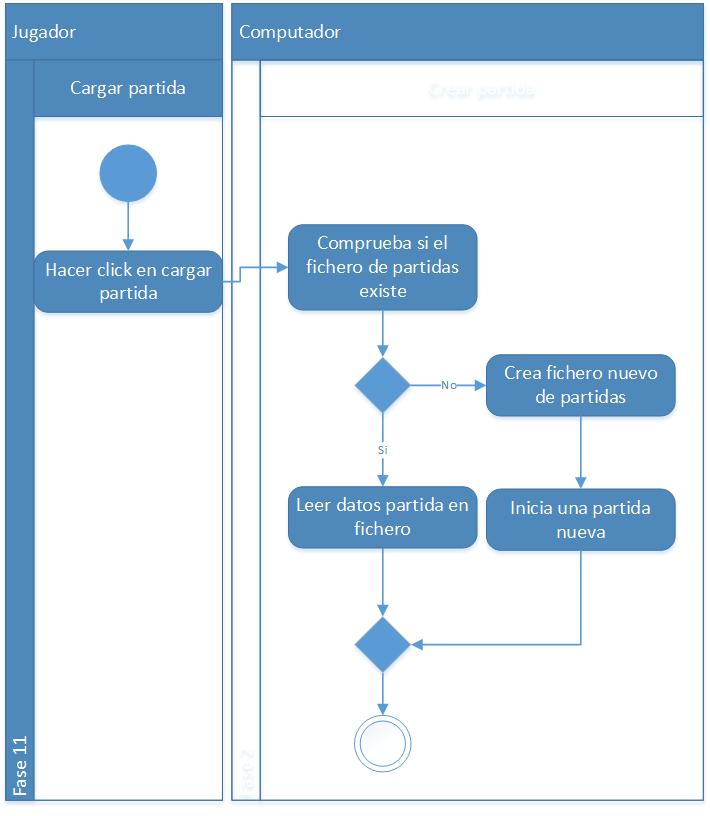
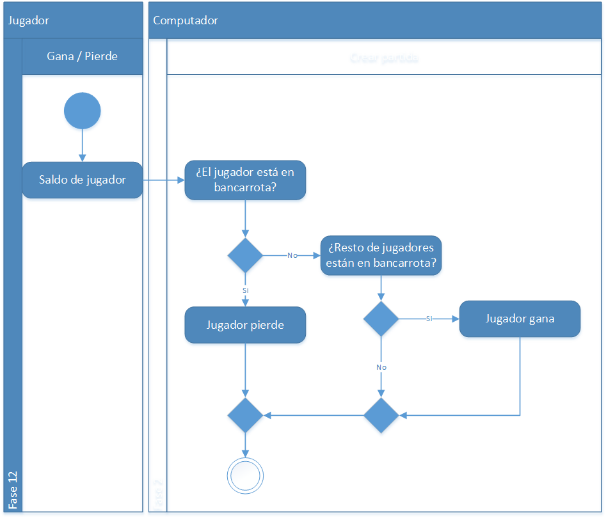
Se incluyen los diagramas de actividad por cada caso de uso que hemos capturado en los requisitos

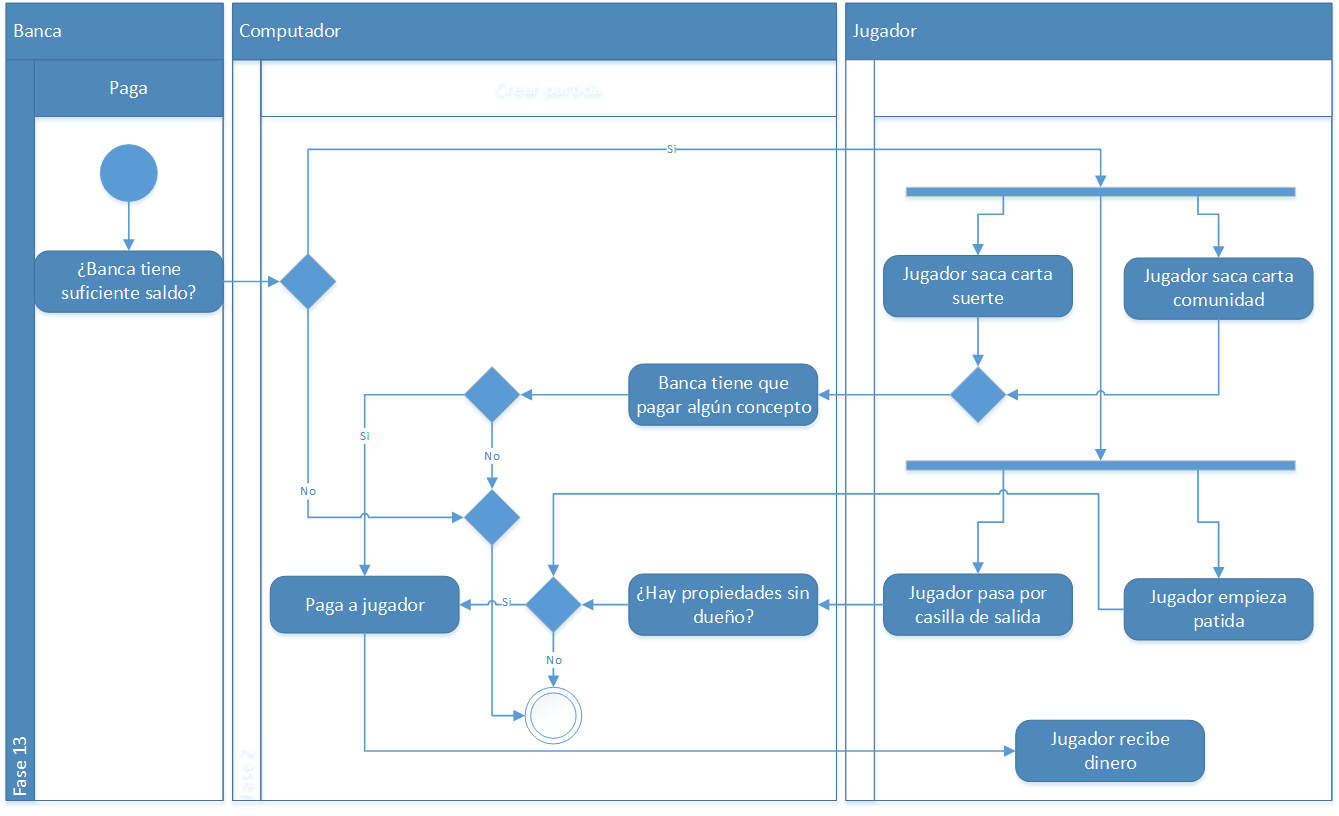
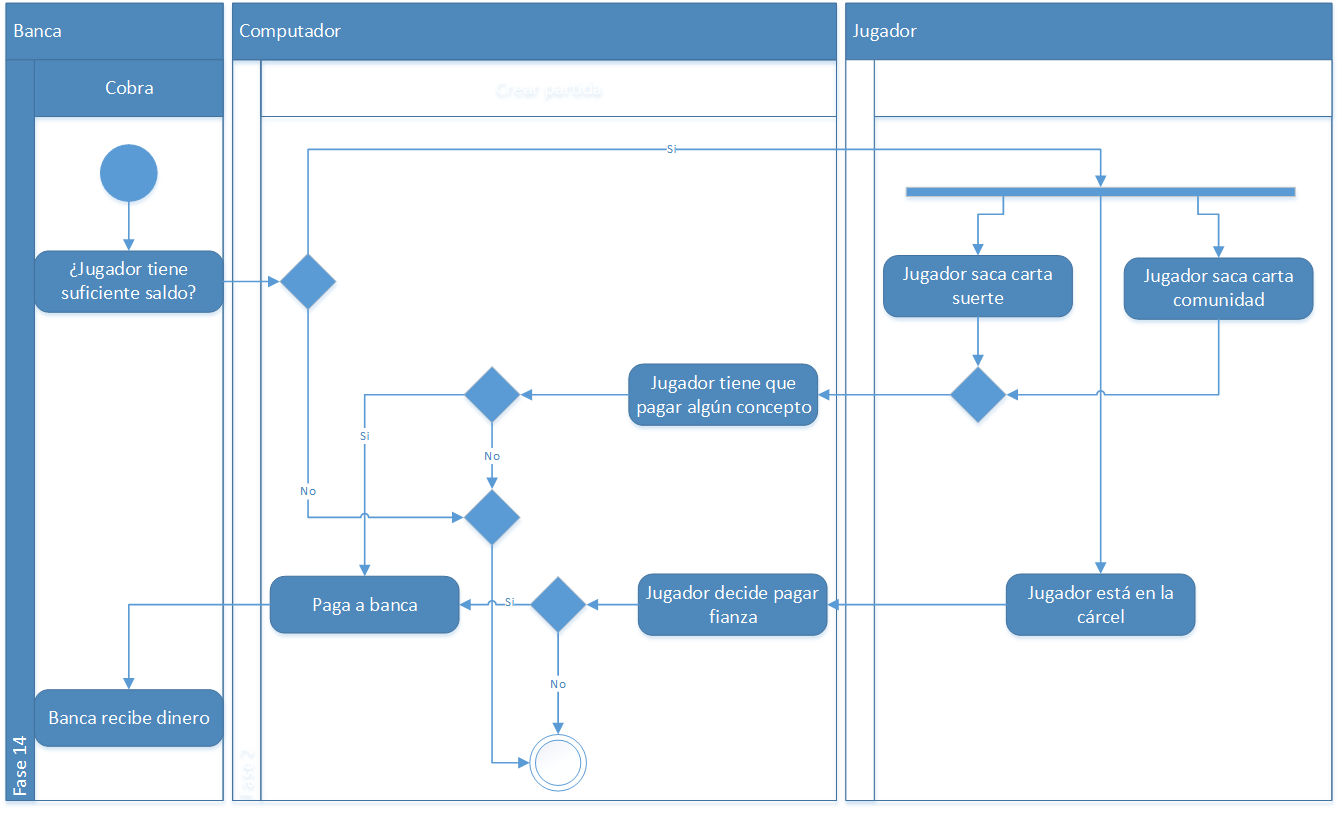
  

# Diagramas de secuencia

Se incluyen los diagramas de secuencia por cada caso de uso que hemos capturado en los requisitos. Como hemos hablado en otras ocasiones, estos diagramas pueden ser modificados a lo largo del desarrollo del proyecto por diferentes razones (captura de nuevos requisitos, errores no previstos, etc)

